December 12, 2017

Machine Learning Approach For User Interface Testing

Hassan Abolhassani

MACHINE LEARNING APPROACH FOR USER INTERFACE TESTING

ABSTRACT

A system and method are disclosed that enable automated testing of a user interface. The testing system includes a machine learning (ML) algorithm to test a user interface. The ML approach includes collecting training samples, creating an ML model and using the model to test the user interface. Training samples are collected by providing the users with diff files with the differences highlighted. A user is provided with options to specify if the differences are acceptable or not. The user classification and other attributes are used to train the model. When a new diff is created it may be fed to the trained network which results in prediction of acceptance or rejection (for example as "ACCEPT DIFF" or "REJECT DIFF") as output of the network. The system eliminates false positives in an automated way and thus reduces time spent by human inspectors to test user interface changes.

BACKGROUND

User interface testing tool may be used to ensure that changes in an application do not adversely affect the look and feel of it. A broad category of testing for this purpose includes screen diffing. Screen diffing is very useful, but is highly sensitive to noise. For example, a pixel shift of a browser page which makes naive pixel to pixel comparison fails when it should not. False positives are bottlenecks to usefulness of the screen diffing approaches.

DESCRIPTION

A system and method are disclosed that enables automated testing of a user interface. The testing system includes a machine learning (ML) algorithm to test a user interface. The method and system includes collection of training samples, creating an ML model and using the model to test the user interface as shown in FIG. 1.

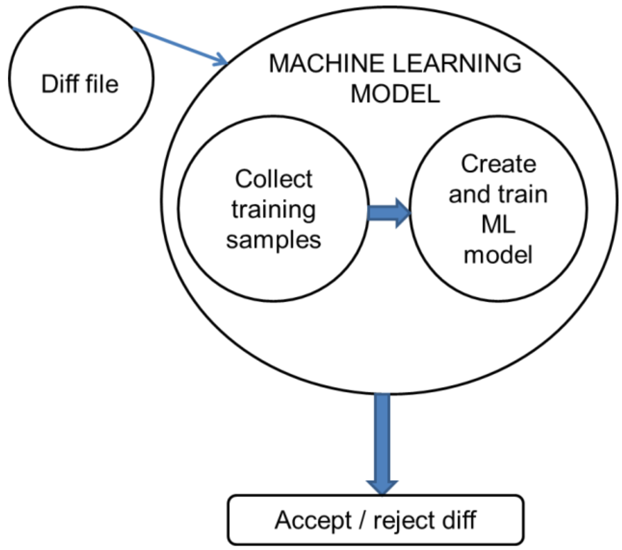

FIG. 1: Machine learning approach for user interface testing

Collection of training samples includes a diff device provided with a graphic user interface (GUI) that shows images produced by golden and under-test systems side by side, with their differences highlighted. A user is provided with options to specify if the differences are acceptable or not. The images with the user decision are stored in a repository that constitutes training samples. Other attributes may also be gathered and stored in the repository as like OS version, browser name or browser version.

Creation of the ML model includes creating a deep learning model based on neural networks by feeding pixels of golden and test images. Other aforementioned categorical attributes are also fed as inputs to the model. User classification of "ACCEPT DIFF" or "REJECT DIFF" is used in training the network.

Based on the trained system, when a diff is created, it may be fed to the trained network which results in prediction of "ACCEPT DIFF" or "REJECT DIFF" as the output of the network. The results are used to filter false positives and hence reduce time spent by human inspectors to resolve diff testing problems. The disclosed system and method may be used to develop screen diffing products.